

文章编号:1001-9014(2005)06-0432-05

# 基于骨架树描述符匹配的物体相似性度量方法

刘文予<sup>1</sup>, 刘俊涛<sup>1,2</sup>

(1. 华中科技大学 电子与信息工程系图像信息处理与智能控制教育部重点实验室, 湖北 武汉 430074;  
2. 军械工程学院 控制工程系, 河北 石家庄 050003)

**摘要:**提出了通过匹配骨架树来度量物体相似度的新方法. 该方法用一种新的物体特征—树描述符来表示一个骨架树, 用搜索树描述符中最长公共子串的方法获得最大同构子树; 用新的模型度量骨架枝的形状相似性, 并把形状特征和拓扑特征的有机的结合起来. 最后, 对骨架有环和骨架上有噪声的情况进行了讨论. 实验结果表明, 本方法计算的相似度具有物体的旋转、大小、平移不变性, 算法时间复杂度为  $O(n^3)$ ,  $n$  为物体骨架枝数, 对多种物体取得了令人满意的结果.

**关键词:**骨架树; 树描述符; 相似性; 匹配; 计算机视觉  
**中图分类号:**TP391.41 **文献标识码:**A

## OBJECTS SIMILARITY MEASURE BASED ON SKELETON TREE DESCRIPTOR MATCHING

LIU Wen-Yu<sup>1</sup>, LIU Jun-Tao<sup>1,2</sup>

(1. Dept. of Electronics & Information Engineering, Key Laboratory of Education Ministry for Image Processing and Intelligent Control, Huazhong University of Science & Technology, Wuhan 430074, China;  
2. Department of Control Engineering, Academy of Ordnance Engineering, Shijiazhuang 050003, China)

**Abstract:** A new object feature, i. e., tree descriptor, was used to represent a skeleton tree. And the maximal isomorphic subtree was obtained by searching the longest same substring in the tree descriptors, then a novel method of objects similarity measuring based on matching skeleton tree was presented. During the matching process, a new model was used to measure the similarity between two skeleton branches, furthermore, the shape feature and topology feature were associated perfectly. The problems of cyclic skeleton and noise on the skeleton were discussed. The similarity computing with our method is invariant under rotation, scaling, translation, and the time complexity of the algorithm is  $O(n^3)$ ,  $n$  is the number of the skeleton branches in the skeleton tree, and the experiments on a variety of objects get satisfying results.

**Key words:** skeleton tree; tree descriptor; similarity; matching; computer vision

### 引言

物体识别在 CAD、集成电路设计、数字化城市、生物医学工程以及军事等领域有着广泛的应用前景<sup>[1,2]</sup>. 自 Blum 首先用骨架<sup>[3]</sup>进行图形匹配以来, 骨架广泛应用于物体识别, 通常先把骨架转化为特征关系图 (Attributed-Relation Graphs, ARG), 然后对特征关系图进行匹配. H. Sundar 等人把物体的骨架用最小生成树的方法转化为骨架图<sup>[4]</sup>, 然后对其进行匹配. K. Siddiqi 等人提出了奇点图 (shock graphs)<sup>[5]</sup>的匹配. 然而, 图的匹配是一个 NPC 问题.

文献[4,5]使用邻接矩阵的特征值之和来度量图中节点的相似性, 进而建立节点间的匹配关系. 文献[6,7]则使用编辑距离来度量奇点图的相似性. D. Geiger 等人则提出用中轴线树 (SA-tree)<sup>[8]</sup>进行物体的匹配. 这些方法计算都相对复杂.

为了更简单、高效的进行物体的辨识, 本文提出用一种新的结构——骨架树, 并用匹配骨架树的方法来识别物体. 与骨架图和奇点图相比, 骨架树的生成规则相对简单, 并且完整的包含了物体的形状特征和拓扑特征.

收稿日期: 2004-12-10, 修回日期: 2005-05-09

Received date: 2004-12-10, revised date: 2005-05-09

基金项目: 国家自然科学基金(60273099)资助项目

作者简介: 刘文予(1963-), 男, 湖南醴陵人, 教授, 主要研究方向: 视觉计算、信息安全.

## 1 骨架树

为了突出物体的拓扑特征,我们把骨架映射到一个树状结构中,称之为骨架树.本文首先考虑骨架没有环的情况,最后再来讨论在骨架有环时如何构造骨架树.

**定义 1** 仅有一个邻接点的骨架点称为端点;有 2 个以上邻接点的骨架点称为分支点;骨架上除分支点和端点外的点称为连接点.如果 2 个骨架点之间仅含有连接点则称这 2 个点是直接相连的.

**定义 2** 直接相连的 2 个骨架点之间的所有连接点的集合称为骨架枝.

如图 1(a) 中的骨架, A、D 为分支点, B、C、E、F 为端点, 其它的点为连接点. 我们把最大内切圆(球)半径的最大且靠近物体重心的骨架点作为根节点. 在图 1(a) 中选择点 A 为骨架树的根节点. 选择了根节点后, 把与根节点直接相连的分支点或端点作为根节点的孩子节点. 图 1(b) 即为(a)中的骨架形成的骨架树, 我们可以看到树中的非叶子节点为骨架上的分支点, 叶子节点为骨架树的端点. 骨架树中的边对应一个骨架枝, 由于这些边是有向的, 故骨架枝也变为一段有向曲线, 其方向为由父节点指向孩子节点. 图 1(b) 中的骨架枝为  $\overrightarrow{AB}$ ,  $\overrightarrow{AC}$ ,  $\overrightarrow{AD}$ ,  $\overrightarrow{DE}$ ,  $\overrightarrow{DF}$ .

为了进行相似性度量, 必须在骨架提取中计算出骨架点的有关特征信息, 如骨架点的分支数、最大半径等, 文献[9]中提出了一种基于距离变换的多尺度连通骨架算法, 可以由粗到精连续提取包含有关特征信息的骨架点, 这为我们骨架树的建立奠定基础.

## 2 树描述符

骨架树将物体的骨架映射到一个二维的树结构中, 为我们识别物体提供了方便. 然而, 树的匹配仍

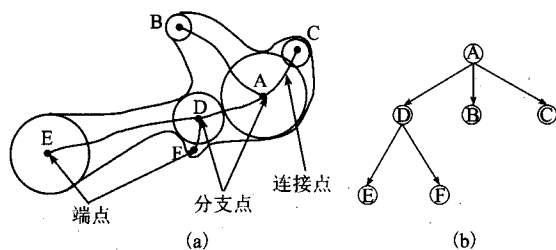


图 1 (a) 骨架以及骨架点的分类 (b) 骨架树  
Fig. 1 (a) Classify of skeleton and skeleton's point  
(b) skeleton tree

然是一个比较复杂的问题, 为此, 我们将骨架树进一步简化为一维的字符串——树描述符. 树描述符是一种新的特征描述结构, 它简洁而完备的反映了骨架树的拓扑结构.

### 2.1 骨架树的深度优先搜索

**定义 3** 对给定的骨架树从根节点开始进行深度优先搜索, 对有多个孩子的节点依据骨架枝在空间中的坐标位置, 按照从后到前、从左到右、从下到上的顺序选择孩子, 这个过程称为骨架树的深度优先搜索.

定义 3 通过比较骨架枝在空间的位置来安排搜索的顺序, 由于骨架枝不可能重合, 一定可以得到唯一的搜索序列. 因此, 我们有下面的定理:

**定理 1** 骨架树的深度优先搜索产生的节点序列是唯一的.

### 2.2 树描述符

**定义 4** 对由骨架树深度优先搜索产生的节点序列中的所有节点用其孩子数替换, 替换后得到的新序列即为树描述符.

例如, 对图 1 中的骨架树进行深度优先搜索产生的节点序列为(A)→(D)→(E)→(F)→(B)→(C), 则树描述符为(3, 2, 0, 0, 0, 0). 树描述符可以看作是一个字符串.

由树描述符重建的树是唯一的, 且重建树和原树具有同样的结构. 由定理 1, 骨架树的深度优先搜索序列是唯一的, 故它的树描述符也是唯一的. 因此, 有下面的定理:

**定理 2** 树和它的描述符是一一对应的.

根据骨架树的定义和定理 2, 可推出树描述符具有下列性质:

**性质 1** 对有  $p$  个节点的树, 其描述符有  $p$  个字符且各项和为  $p-1$ .

**性质 2** 树的一个子树的描述符是该树的描述符的一个连续子串, 并且这个子串的各项和等于子树节点数减 1, 即该子串也满足性质 1.

事实上, 性质 2 给出了树描述符的任意子串是否是一个子树描述符的判定, 即树描述符中的子串, 如果满足性质 1 则它必定是一个子树的描述符. 例如, 图 1 中树的描述符为(3, 2, 0, 0, 0, 0), 子串(2, 0, 0)满足性质 1, 则它是一个子树的描述符; 而子串(3, 2, 0, 0)不满足性质 1, 所以它不是一个子树的描述符.

### 2.3 搜索最大同构子树

所谓最大同构子树指的是结构相同且包含节点

最多的子树,由定理 2 易知同构的树必有相同的树描述符. 根据性质 2,我们可以把求最大同构子树的问题转化为在 2 个树描述符中搜索最长公共子串的问题,搜索到的子串必须满足性质 1,否则找到的公共子串不是子树的描述符.

据此,可以得到搜索 2 个骨架树中最大同构子树的方法. 对于 2 个骨架树  $T_1, T_2$ , 先分别求它们的描述符  $s_1, s_2$ , 然后求  $s_1, s_2$  中满足性质 1 的最长公共子串,即找到了 2 个骨架树的最大同构子树.

### 3 骨架树的匹配

#### 3.1 算法概述

本算法的基本思想是在 2 个骨架树中搜索最大同构子树,建立 2 个同构子树骨架枝之间的匹配关系;从原树中删除已匹配的子树,对剩下的部分进行同样的操作,直到其中的一个骨架树为空;最后,计算匹配的骨架枝的相似度,并将这些相似度的和作为骨架树的相似度. 需解决的问题是如何计算两个骨架枝的相似度,如何建立同构骨架树中骨架枝的匹配关系,下面分别加以说明.

#### 3.2 骨架枝的相似度模型

设需计算相似度的 2 个骨架枝为  $bt_1 = (C_1(s), R_1(s)), bt_2 = (C_2(s), R_2(s))$ , 其中  $s$  为曲线方程的弧长参数.  $C_1(s), C_2(s)$  为骨架枝的曲线方程.  $R_1(s), R_2(s)$  为最大内切球(圆)半径.

设  $bt_1, bt_2$  的长度分别为  $L_1, L_2$ , 不妨设  $L_1 > L_2$ . 对齐操作对弧长较短的骨架枝进行拉伸,使它的长度与较长的骨架枝相等,进行对齐后 2 个骨架枝有相同的长度  $L_1$ , 可以用同一个参数来表示. 相似度定义为平均弯曲度差异与平均区域差异的加权和. 平均弯曲度差异可用下式计算:

$$k = \frac{1}{L_1} \int_0^{L_1} |\kappa_1(s) - \kappa_2(s)| ds. \quad (1)$$

其中  $\kappa_1(s)$  和  $\kappa_2(s)$  分别为对齐后骨架枝曲线的曲率. 平均区域差异用来表示物体轮廓上的差异,可以用下式计算:

$$b = \frac{1}{L_1} \int_0^{L_1} |R_1(s) - \bar{R}_2(s)| ds. \quad (2)$$

$\bar{R}_2(s)$  为对齐后最大内切球半径. 由上面的计算可得到骨架枝的相似度:

$$\begin{aligned} s(bt_1, bt_2) &= C - \frac{L_1}{L_2} (q_1 \times k + q_2 \times b) \\ &= C - \frac{1}{L_2} \int_0^{L_1} q_1 |\kappa_1(s) - \kappa_2(s)| \end{aligned}$$

$$+ q_2 |R_1(s) - \bar{R}_2(s)| ds. \quad (3)$$

其中  $q_1, q_2 \in [0, 1], q_1 + q_2 = 1$ . 常数  $q_1, q_2$  表明了平均弯曲度差异与平均区域差异的重要程度;系数  $L_1/L_2$  将拉伸的代价计入相似度中,为了符合相似度的实际意义而取了负号;加上一个合适大小的正常数  $C$ , 仅仅为了使其变为正值. 这样处理后,数值越大则表示越相似,符合相似度的实际意义.

#### 3.3 骨架树的相似度计算

为了计算 2 个骨架树的相似度,需要先建立骨架枝之间的匹配关系,计算匹配的骨架枝的相似度之和,以此作为整个骨架的相似度,对于不能找到匹配对象的骨架枝,则不予处理.

设需计算相似度的骨架树为  $T_1, T_2$ , 分别有  $k, n$  个骨架枝,它们的前  $m$  个骨架枝间已建立了匹配关系,根据上面的分析,它们的相似度可以计算如下:

$$\text{sim}(T_1, T_2) = \frac{m}{\text{MAX}(k, n)} \sum_{i=1}^m s(T_1(i), T_2(i)). \quad (4)$$

其中,  $T(i)$  表示骨架树中第  $i$  个骨架枝,骨架枝之间的相似度  $s(T_1(i), T_2(i))$  是由式(3)得到的.

##### 3.3.1 同构骨架的匹配

**定义 5** 设  $u, v$  分别是属于 2 个同构树中的节点,如果以  $u, v$  为根的子树是同构的且它们的父节点间已经建立了匹配关系,则在  $u, v$  之间可以建立匹配关系( $u \sim v$ ),称为可行匹配.

**定义 6** 设  $u_0, v_0$  分别是属于 2 个同构树中的节点,且  $u_0, v_0$  之间已建立匹配关系( $u_0 \sim v_0$ ),则在它们的直接孩子节点间建立的某种一一对应的可行匹配关系称为由( $u_0 \sim v_0$ )导出的匹配, ( $u_0 \sim v_0$ )称为导出者. 由这种导出匹配会产生一些匹配的边,匹配的边之间的相似度之和称为导出相似度.

**算法 1:** 建立同构骨架树  $T_1, T_2$  的边之间的匹配关系,在  $T_1, T_2$  的根节点之间建立匹配关系  $m_1 = (1 \sim 1)$  (根节点编号为 1);  $s_1 = 0$ ;

1. 将  $m_1 = (1 \sim 1)$  加入可行匹配集合 FSM 中;  $i = 2$ ;

2. 找出由 FSM 中的匹配关系导出的且不在 FSM 中的所有的匹配关系  $\{tm_j\}$ , 并计算它们的导出相似度  $\{ts_j\}$ , 设这些匹配关系的导出者分别是 FSM 中的第  $\{p_j\}$  个匹配关系. 设当  $j = k$  时,  $ts_j + S_{pk}$  达到最大值. 则把  $m_i = tm_k$  加入 FSM 中并记下它的导出者;  $S_i = ts_k + S_{pk}; i++$ ;

3. 当 FSM 中包含了所有叶子节点的匹配关系

时结束,否则返回 3.

算法 1 使用了动态规划的方法,能找到最优的匹配.最后,取出  $\{S_i\}$  中的最大数值就得到两个同构骨架树的相似度.

### 3.3.2 异构骨架的匹配

设需计算相似度的骨架树为  $T_1$ 、 $T_2$ , 分别有  $L_1$ 、 $L_2$  条边, 设  $L_1 > L_2$ . 下面的算法计算它们的相似度.

**算法 2:** 计算异构骨架树的相似度

1.  $d_1 = \text{Desc}(T_1)$ ,  $d_2 = \text{Desc}(T_2)$ ;  $\text{sim} = 0$ ;
2. 搜索  $d_1, d_2$  中的最长公共子串, 设取得的子串的首字符在  $d_1, d_2$  中分别是第  $i$  个和第  $j$  个字符, 用算法 1 计算  $T_1, T_2$  中分别以  $i, j$  为根的子树的相似度  $S_0$ , 若  $i, j$  都不是根节点则还要计算  $T_1$  中以第  $i$  个节点为终点的边和  $T_2$  中以  $j$  个节点为终点的边之间的相似度  $s_0$ ; 若最长公共子串有多个, 选择使  $S_0 + s_0$  的数值达到最大值  $S$  的一对子树, 建立边的匹配关系. 若找不到相同的子串则计算结束;
3.  $\text{sim} = \text{sim} + S$ ;
4. 分别从  $T_1, T_2$  中删除已经建立匹配关系的边, 形成新的  $T_1, T_2$ ;
5.  $d_1 = \text{Desc}(T_1)$ ,  $d_2 = \text{Desc}(T_2)$ ;
6. 如果  $d_1, d_2$  的长度都  $> 1$ , 则返回 2, 否则结束, 两个骨架树的相似度为  $\text{sim} \times L_2 / L_1$ .

其中  $\text{Desc}(T)$  表示求骨架树  $T$  的描述符的操作. 算法 2 可以得到任意骨架树的相似度, 它的核心是对同构子树的搜索, 这些运算因为树描述符的引入而得到简化.

算法 2 首先获得 2 个骨架树的树描述符, 搜索树描述符中的最长公共子串, 而最长公共子串的首字符在树描述符中的位置即为最大同构子树的根节点的编号. 再用算法 1 计算最大同构子树的相似度.

另外, 还需计算以同构子树的根节点为终点的边的相似度. 此时, 我们删除原骨架树中已经建立匹配关系的边, 形成新的骨架树进行同样的操作.

对于算法 2, 主要的计算集中在第 2 步, 即搜索最大同构子树. 对于 2 个分别有  $n_1, n_2$  个边的骨架树, 算法 2 的第 2 步至多执行  $n = \text{MIN}(n_1, n_2)$  次, 搜索最大同构子树的时间复杂度为  $o(n_1 \times n_2)$ . 整个骨架树匹配的算法可以在  $o(n^3)$  的时间内完成.

## 4 实验结果及讨论

我们对大量的物体进行了实验, 表 1 是实验中的部分物体的图像及骨架, 骨架的计算由文献[9]得到. 表 2 是根据算法 2 计算的相似度, 表中的数据为所在行、列 2 个物体的相似度经过归一化处理后的结果, 所谓归一化处理是将一行中的相似度除以该行的最大数值. 处理后相似度在  $[0, 1]$  之间, 数值越大表示越相似, 0 表示完全不相似, 1 表示极其相似. 表 2 中带下画线的数值是该行物体除去与本身相比外最大的数值, 即与除本身外最相似的物体的

表 1 用于实验的图像及其骨架

Table 1 The images and its skeleton for experiments

























编号	1	2	3	4	5	6
原始图像						
骨架						
编号	7	8	9	10	11	12
原始图像						
骨架						

表 2 部分实验结果

Table 2 The results of experiment

物体编号	1	2	3	4	5	6	7	8	9	10	11	12
1	1.000	0.802	0.408	0.318	0.507	0.605	0.032	0.028	0.083	0.300	0.264	0.305
2	0.891	1.000	0.492	0.446	0.504	0.531	0.038	0.032	0.164	0.420	0.360	0.403
3	0.584	0.632	1.000	0.957	0.388	0.370	0.075	0.076	0.754	0.625	0.553	0.558
4	0.454	0.573	0.957	1.000	0.375	0.396	0.058	0.067	0.830	0.439	0.615	0.618
5	0.298	0.267	0.160	0.154	1.000	0.739	0.013	0.013	0.177	0.136	0.110	0.114
6	0.403	0.319	0.173	0.185	0.838	1.000	0.016	0.013	0.098	0.148	0.156	0.154
7	0.160	0.173	0.263	0.202	0.112	0.123	1.000	0.958	0.234	0.223	0.295	0.266
8	0.142	0.145	0.265	0.235	0.108	0.099	0.958	1.000	0.240	0.236	0.197	0.201
9	0.104	0.185	0.660	0.726	0.377	0.184	0.059	0.060	1.000	0.965	0.311	0.279
10	0.375	0.472	0.547	0.384	0.288	0.277	0.056	0.058	0.965	1.000	0.301	0.260
11	0.440	0.540	0.646	0.718	0.313	0.389	0.098	0.066	0.415	0.402	1.000	0.963
12	0.508	0.604	0.650	0.721	0.321	0.385	0.089	0.067	0.372	0.346	0.963	1.000

相似度. 可以看到, 实验获得了令人满意的结果.

表中  $C = 100$ ;  $q_1 = 0.2$ ;  $q_2 = 0.8$ ; 每一行中的相似度已经进行了归一化, 以下同.

需要说明的是关于  $q_1$ 、 $q_2$  的选择. 表中的结果是在  $q_1 = 0.2$ 、 $q_2 = 0.8$  的情况下计算的. 由于这些图像的骨架支曲线都比较平直, 所以平均弯曲度差异比较小, 而平均区域差异比较明显, 所以加重了它的权重, 而减小平均弯曲度的权重. 可以看到  $q_1$ 、 $q_2$  的选择使本文的方法具有一定的灵活性.

本文提出的方法同样能够处理三维物体, 由于三维物体的骨架提取仍然是一个棘手的问题, 所以我们仅对那些能够方便获得骨架的三维物体进行了实验. 表 3 中显示了 4 个三维物体和它们之间的相似度. 我们的方法对三维物体也获得了很好的区分.

对于环形的骨架, 按前面的方法不能得到 1 个树形的结构, 因为至少有 1 个节点有 2 个以上父节点. 对这样的节点进行处理, 即可得到骨架树. 设节点  $P$  有  $n(n > 1)$  个父节点, 将  $P$  复制  $n$  个副本:  $P_1$ ,  $P_2, \dots, P_n$ , 把  $P_i$  与  $P$  的第  $i$  个父节点相连接, 同时,  $P_i$  继承  $P$  的所有孩子节点, 其它的  $P_i$  则设置为叶子节点. 处理完所有这样的节点后, 就得到 1 个骨架树.

## 5 结语

本文提出了一种新的物体相似度计算方法, 与


















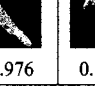
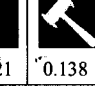

现有的物体识别的方法相比, 该方法通过简单的规则生成骨架树, 完整的表达了物体的形状和拓扑特征; 用新的特征结构——树描述符来描述骨架树的拓扑结构, 避免了复杂的运算, 基于树描述符的同构子树搜索方法简单而快速; 骨架枝相似度模型中, 对平均弯曲度差异与平均区域差异权重的选择, 使得该模型具有灵活性; 在骨架树匹配的过程中把对形状特征和拓扑特征的比较有机的结合起来, 使得匹配能够获得准确的结果. 本方法计算的相似度具有物体的旋转、大小、平移不变性, 算法时间复杂度为  $o(n^3)$ , 这里  $n$  为物体骨架的分支数. 实验结果表明, 对多种物体取得了令人满意的结果.

## REFERENCES

- [1] Choi H I, Choi S W, Moon H P, et al. New algorithm for medial axis transform of plane domain [J]. *Graphical Models and Image Processing*, 1997, **59**(6): 463—483.
- [2] Dimitrov P, Carlos P, Kaleem S. Robust and efficient skeletal graphs [C]. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE Press*, Hilton Head Island, South Carolina, USA, 2000: 417—423.
- [3] Blum H. Biological shape and visual science: Part I [J]. *Journal of Theoretical Biology*, 1973, **38**: 205—287.
- [4] Sundar H, Silver D, Gagvani N, et al. *Skeleton Based Shape Matching and Retrieval* [M]. In: Proc. Shape Modeling International, Ed. M.-S. Kim. *IEEE Press*. Seoul Korea, 2003: 130—290.
- [5] Kaleem Siddiqi, Bnejamin B. Kimia. A shock grammar for recognition [C]. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, IEEE Press*, San Francisco, USA, 1996: 507—513.
- [6] Srikanta Tirthapura, Daniel Sharvit, Philip Klein, et al. *Indexing Based on Edit-Distance Matching of Shape Graphs* [M]. In: *Proceedings of SPIE*, Ed: Kuo, Chung-chieh Jay, Bellingham, Wash: USA, 1998, **3527**: 25—36.
- [7] Benjamin B Kimia. Recognition of shapes by editing shock graphs [C]. In: *Proceedings of the international conference on computer vision. IEEE Press*, Vancouver: Canada, 2001, **1**: 755—762.
- [8] Davi Geiger, Tyng-Luh Liu, Robert V Kohn. Representation and self-similarity of shapes [J]. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2003, **25**(1): 86—99.
- [9] DING Yi, LIU Wen-Yu. A hierarchical connected skeletonization algorithm based on the distance transform [J]. *Journal of Infrared and Millimeter Waves* (丁颐, 刘文予. 基于距离变换的多尺度连通骨架算法. 红外与毫米波学报), 2005, **24**(4): 281—285.

表 3 三维物体的相似度

Table 3 Similarity among the 3D images

查询物体	4 个最相似的物体			
	1	2	3	4
	 1.000	 0.414	 0.108	 0.107
	 1.000	 0.178	 0.020	 0.020
	 1.000	 0.976	 0.325	 0.138
	 1.000	 0.976	 0.321	 0.138